EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	20	717/128.ccls. and (trac$3 with count$3 with value) and trigger$3 and buffer	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/12 12:21
L2	150	717/124-129.ccls. and (trac$3 and count$3 and trigger$3 and buffer)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/12 12:23
L3	137	714/45.ccls. and (trac$3 and count$3 and trigger$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/12 12:24
L4	150	714/38.ccls. and (trac$3 and count$3 and trigger$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/12 12:24
L5	120	712/227.ccls. and (trac$3 and count$3 and trigger$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/12 12:24
S1	50	("20040123084" "20050060521" "6223338" "6223338" "6240529" "5574937" "5944841" "5996092" "7062684" "5560036" "6145123" "20030204707" "20050177703" "20050177819" "4435759" "5648836" "5832318" "20030051231" "5809293" "6112019" "5272424" "5764970" "5881260" "6003128" "6311261" "6397379" "4293950" "6233678" "6351844" "5901283" "20040030962" "6026503" "4879646" "5724505" "5751942" "5787276" "6023561" "6173395" "6507921" "6826747" "20040117690" "20060085688" "5257358" "5289587" "4381563" "5944806" "6687857" "5771410" "6253338" "5802346").pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/10 19:05
S2	31	"6032268"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/11 14:16

EAST Search History

S3	50	("20040123084" "20050060521" "6223338" "6223338" "6240529" "5574937" "5944841" "5996092" "7062684" "5560036" "6145123" "20030204707" "20050177703" "20050177819" "4435759" "5648836" "5832318" "20030051231" "5809293" "6112019" "5272424" "5764970" "5881260" "6003128" "6311261" "6397379" "4293950" "6233678" "6351844" "5901283" "20040030962" "6026503" "4879646" "5724505" "5751942" "5787276" "6023561" "6173395" "6507921" "6826747" "20040117690" "20060085688" "5257358" "5289587" "4381563" "5944806" "6687857" "5771410" "6253338" "5802346").pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/11 14:29
S4	23	S3 and (trac$3 and trigger$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/11 14:50
S5	13	S3 and (trac$3 and trigger$3 and ((real run$3) adj1 time))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/11 14:51

1 Real-time shading

Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi Rost

August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: pdf(7.39 MB) Additional Information: full citation, abstract

Real-time procedural shading was once seen as a distant dream. When the first version of this course was offered four years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combining the effects of tens to hundreds of rendering passes. Today, almost every new computer comes with graphics hardware capable of interactively executing shaders of thousands to tens of thousands of instructions. This course has been redesigned to address today's real-time shading capabili ...

2 The embedded machine: predictable, portable real-time code

Thomas A. Henzinger, Christoph M. Kirsch

May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation PLDI '02**, Volume 37 Issue 5

Publisher: ACM Press

Full text available: pdf(223.85 KB) Additional Information: full citation, abstract, references, citings, index terms

The Embedded Machine is a virtual machine that mediates in real time the interaction between software processes and physical processes. It separates the compilation of embedded programs into two phases. The first, platform-independent compiler phase generates E code (code executed by the Embedded Machine), which supervises the timing ---not the scheduling--- of application tasks relative to external events, such as clock ticks and sensor interrupts. E~code is portable and exhibits, given an inpu ...

Keywords: real time, virtual machine

3 An efficient single-pass trace compression technique utilizing instruction streams

Aleksandar Milenković, Milena Milenković

January 2007 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 17 Issue 1

Publisher: ACM Press

Full text available: Additional Information:

BEST AVAILABLE COPY

pdf(848.21 KB) full citation, abstract, references, index terms

Trace-driven simulations have been widely used in computer architecture for quantitative evaluations of new ideas and design prototypes. Efficient trace compression and fast decompression are crucial for contemporary workloads, as representative benchmarks grow in size and number. This article presents Stream-Based Compression (SBC), a novel technique for single-pass compression of address traces. The SBC technique compresses both instruction and data addresses by associating them with a particu ...

Keywords: Instruction and data traces, instruction streams, trace compression

4 Hardware monitoring of real-time aerospace computer systems

D. R. Partridge, R. E. Card

March 1976 **Proceedings of the 1976 ACM SIGMETRICS conference on Computer performance modeling measurement and evaluation SIGMETRICS '76**

Publisher: ACM Press

Full text available: pdf(1.00 MB) Additional Information: full citation, abstract, index terms

Hardware monitoring has proven to be a useful means for measuring the performance of computer systems generally, and is particularly attractive for use on real-time systems due to its attribute of non-interference with system operation. This technique is uniquely able to quantify precisely the interactions between hardware and software, which must be completely understood in these systems. In this paper, we report the application of a commercially-developed hardware monitor to two real-time ...

5 A real-time microprocessor debugging technique

Charles R. Hill

March 1983 **ACM SIGPLAN Notices , ACM SIGSOFT Software Engineering Notes , Proceedings of the symposium on High-level debugging SIGSOFT '83,** Volume 18 , 8 Issue 8 , 4

Publisher: ACM Press

Full text available: pdf(380.29 KB) Additional Information: full citation, abstract, references

This note describes RED, a remotely executed debugger capable of generating a real-time source level trace history of a high level language program executing on a microprocessor. The trace history consists of a display of the source statements of each basic block executed, annotated by the time at which execution of that block began. Basic blocks are traced rather than statements to reduce sampling bandwidth requirements while still retaining the ability to record the essential logical flow of p ...

6 LIFT: A Low-Overhead Practical Information Flow Tracking System for Detecting Security Attacks

Feng Qin, Cheng Wang, Zhenmin Li, Ho-seop Kim, Yuanyuan Zhou, Youfeng Wu

December 2006 **Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture MICRO '06**

Publisher: IEEE Computer Society

Full text available: pdf(254.69 KB) Additional Information: full citation, abstract

Computer security is severely threatened by software vulnerabilities. Prior work shows that information flow tracking (also referred to as taint analysis) is a promising technique to detect a wide range of security attacks. However, current information flow tracking systems are not very practical, because they either require program annotations, source code, non-trivial hardware extensions, or incur prohibitive runtime overheads. This paper proposes a low overhead, software-only information flow ...

7 Collaborative operating system and compiler power management for real-time

BEST AVAILABLE COPY

applications
Nevine AbouGhazaleh, Daniel Mossé, Bruce R. Childers, Rami Melhem
February 2006 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 5 Issue 1
Publisher: ACM Press
Full text available: pdf(733.43 KB) Additional Information: full citation, abstract, references, index terms

Managing energy consumption has become vitally important to battery-operated portable and embedded systems. *Dynamic voltage scaling* (DVS) reduces the processor's dynamic power consumption quadratically at the expense of linearly decreasing the performance. When reducing energy with DVS for real-time systems, one must consider the performance penalty to ensure that deadlines can be met. In this paper, we introduce a novel collaborative approach between the compiler and the operating system ...

Keywords: Real-time, collaborative OS and compiler, dynamic voltage scaling, power-management, voltage scaling points placement

8 Cache Memories
Alan Jay Smith
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3
Publisher: ACM Press
Full text available: pdf(4.61 MB) Additional Information: full citation, references, citings, index terms

9 GPGPU: general purpose computation on graphics hardware
David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(63.03 MB) Additional Information: full citation, abstract, citings

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

10 Compiling real-time programs into schedulable code
Seongsoo Hong, Richard Gerber
June 1993 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 conference on Programming language design and implementation PLDI '93**, Volume 28 Issue 6
Publisher: ACM Press
Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

We present a programming language with first-class timing constructs, whose semantics is based on time-constrained relationships between observable events. Since a system specification postulates timing relationships between events, realizing the specification in a program becomes a more straightforward process. Using these constraints, as well as those imposed by data and control flow properties, our objective is to transform the code so that its worst-case execution time is con ...

11 Collecting whole-system reference traces of multiprogrammed and multithreaded

BEST AVAILABLE COPY

workloads
Scott F. Kaplan
January 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 4th international workshop on Software and performance WOSP '04**, Volume 29 Issue 1
Publisher: ACM Press
Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, citings

The simulated evaluation of memory management policies relies on *reference traces*--logs of memory operations performed by running processes. No existing approach to reference trace collection is applicable to a complete system, including the kernel and all processes. Specifically, none gather sufficient information for simulating the virtual memory management, the filesystem cache management, and the scheduling of a multiprogrammed, multithreaded workload. Existing trace collectors are al ...

12 Constructing instruction traces from cache-filtered address traces (CITCAT)
Charlton D. Rose, J. Kelly Flanagan
December 1996 **ACM SIGARCH Computer Architecture News**, Volume 24 Issue 5
Publisher: ACM Press
Full text available: pdf(595.54 KB) Additional Information: full citation, abstract, citings, index terms

Instruction traces are useful tools for studying many aspects of computer systems, but they are difficult to gather without perturbing the systems being traced. In the past, researchers have collected instruction traces through various techniques, including single-stepping, instruction inlining, hardware monitoring, and processor simulation. These approaches, however, fail to produce accurate traces because they interfere with the processor's normal execution.Because processors are deterministic ...

13 Automated discovery of scoped memory regions for real-time Java
Morgan Deters, Ron K. Cytron
June 2002 **ACM SIGPLAN Notices , Proceedings of the 3rd international symposium on Memory management ISMM '02**, Volume 38 Issue 2 supplement
Publisher: ACM Press
Full text available: pdf(227.49 KB) Additional Information: full citation, abstract, references, citings, index terms

Advances in operating systems and languages have brought the ideal of reasonably-bounded execution time closer to developers who need such assurances for real-time and embedded systems applications. Recently, extensions to the Java libraries and virtual machine have been proposed in an emerging standard, which provides for specification of release times, execution costs, and deadlines for a restricted class of threads. To use such features, the code executing in the thread must never reference s ...

Keywords: garbage collection, memory management, real-time Java, regions, trace-based analysis

14 Integrating real-time and partial-order information in event-data displays
David J. Taylor, Michael H. Coffin
October 1994 **Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research CASCON '94**
Publisher: IBM Press
Full text available: pdf(149.09 KB) Additional Information: full citation, abstract, references, citings, index terms

The events occurring in the execution of a distributed or parallel application are related by a partial, rather than a total, order. We have developed prototype software that collects such events during program execution and produces a graphical display consistent with

BEST AVAILABLE COPY

the partial order. Such a display can be very helpful in understanding and debugging distributed and parallel applications. However, using only partial-order information does not allow the performance characteristics of an applica ...

15 OS: Architectural support for real-time task scheduling in SMT processors

Francisco J. Cazorla, Peter M. W. Knijnenburg, Rizos Sakellariou, Enrique Fernández, Alex Ramirez, Mateo Valero

September 2005 **Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems CASES '05**

Publisher: ACM Press

Full text available: pdf(279.85 KB) Additional Information: full citation, abstract, references, index terms

In Simultaneous Multithreaded (SMT) architectures most hardware resources are shared between threads. This provides a good cost/performance trade-off which renders these architectures suitable for use in embedded systems. However, since threads share many resources, they also interfere with each other. As a result, execution times of applications become highly unpredictable and dependent on the context in which an application is executed. Obviously, this poses problems if an SMT is to be used in ...

Keywords: ILP, SMT, multithreading, performance predictability, real time, scheduling, thread-level parallelism

16 Virtual simple architecture (VISA): exceeding the complexity limit in safe real-time systems

Aravindh Anantaraman, Kiran Seth, Kaustubh Patil, Eric Rotenberg, Frank Mueller

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture ISCA '03**, Volume 31 Issue 2

Publisher: ACM Press

Full text available: pdf(147.00 KB) Additional Information: full citation, abstract, references, citings

Meeting deadlines is a key requirement in safe realtime systems. Worst-case execution times (WCET) of tasks are needed for safe planning. Contemporary worst-case timing analysis tools can safely and tightly bound execution time on in-order single-issue pipelines with caches and static branch prediction. However, this simple pipeline appears to be a complexity limit, due to the need for analyzability. This excludes a whole class of high-performance processors from many embedded systems.We reconci ...

17 Compositional static instruction cache simulation

Kaustubh Patil, Kiran Seth, Frank Mueller

June 2004 **ACM SIGPLAN Notices , Proceedings of the 2004 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems LCTES '04**, Volume 39 Issue 7

Publisher: ACM Press

Full text available: pdf(192.78 KB) Additional Information: full citation, abstract, references, index terms

Scheduling in hard real-time systems requires a priori knowledge of worst-case execution times (WCET). Obtaining the WCET of a task is a difficult problem. Static timing analysis techniques approach this problem via path analysis, pipeline simulation and cache simulation to derive safe WCET bounds. But such analysis has traditionally been constrained to only small programs due to the complexity of simulation, most notably the complexity of static cache simulation, which requires inter-procedural ...

Keywords: caches, real-time systems, scheduling, worst-case execution time

BEST AVAILABLE COPY

18 The effects of workspace awareness support on the usability of real-time distributed groupware

Carl Gutwin, Saul Greenberg

September 1999 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 6 Issue 3

Publisher: ACM Press

Full text available: pdf(494.20 KB) Additional Information: full citation, references, citings, index terms, review

Keywords: computer-supported cooperative work, real-time distibuted groupware, usability, workspace awareness

19 Exploiting perception in high-fidelity virtual environments: Exploiting perception in high-fidelity virtual environments

Additional presentations from the 24th course are available on the citation page

Mashhuda Glencross, Alan G. Chalmers, Ming C. Lin, Miguel A. Otaduy, Diego Gutierrez

July 2006 **ACM SIGGRAPH 2006 Courses SIGGRAPH '06**

Publisher: ACM Press

Full text available: pdf(5.07 MB) mov(68:6 MIN) Additional Information: full citation, abstract, references

The objective of this course is to provide an introduction to the issues that must be considered when building high-fidelity 3D engaging shared virtual environments. The principles of human perception guide important development of algorithms and techniques in collaboration, graphical, auditory, and haptic rendering. We aim to show how human perception is exploited to achieve realism in high fidelity environments within the constraints of available finite computational resources.In this course w ...

Keywords: collaborative environments, haptics, high-fidelity rendering, human-computer interaction, multi-user, networked applications, perception, virtual reality

20 Real-time volume graphics

Klaus Engel, Markus Hadwiger, Joe M. Kniss, Aaron E. Lefohn, Christof Rezk Salama, Daniel Weiskopf

August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: pdf(7.63 MB) Additional Information: full citation, abstract

The tremendous evolution of programmable graphics hardware has made high-quality real-time volume graphics a reality. In addition to the traditional application of rendering volume data in scientific visualization, the interest in applying these techniques for real-time rendering of atmospheric phenomena and participating media such as fire, smoke, and clouds is growing rapidly. This course covers both applications in scientific visualization, e.g., medical volume data, and real-time rendering, ...

BEST AVAILABLE COPY

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player